

Spectral projected gradient method for generalized tensor eigenvalue complementarity problem

Gaohang Yu ^{*} Yisheng Song [†] Yi Xu [‡] Zefeng Yu ^{*}

January 11, 2016

Abstract

This paper looks at the tensor eigenvalue complementarity problem (TEiCP) which arises from the stability analysis of finite dimensional mechanical systems and is closely related to the optimality conditions for polynomial optimization. We investigate two monotone ascent spectral projected gradient (SPG) methods for TEiCP. We also present a shifted scaling-and-projection algorithm (SPA), which is a great improvement of the original SPA method proposed by Ling, He and Qi [Comput. Optim. Appl., DOI 10.1007/s10589-015-9767-z]. Numerical comparisons with some existed gradient methods in the literature are reported to illustrate the efficiency of the proposed methods.

Keywords: Tensor, Pareto Eigenvalue, Pareto Eigenvector, Projected gradient method, Eigenvalue Complementarity Problem.

1 Introduction

A m th-order n -dimensional real tensor \mathcal{A} consists of n^m entries in real numbers:

$$\mathcal{A} = (a_{i_1 i_2 \dots i_m}), \quad a_{i_1 i_2 \dots i_m} \in \mathbb{R}, \quad \text{for any } i_1, i_2, \dots, i_m \in [n],$$

where $[n] = \{1, 2, \dots, n\}$. Denote the set of all real m th-order n -dimensional tensors by $\mathbb{T}^{[m, n]}$. \mathcal{A} is called *symmetric* if the value of $a_{i_1 i_2 \dots i_m}$ is invariant under any permutation

^{*}School of Mathematics and Computer Sciences, Gannan Normal University, Ganzhou, 341000, China. E-mail: maghyu@163.com

[†]School of Mathematics and Information Science and Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control, Henan Normal University, China. E-mail: songyisheng1@gmail.com

[‡]Department of Mathematics, Southeast University, China. E-mail: yi.xu1983@gmail.com

of its indices i_1, i_2, \dots, i_m . Denote the set of all real symmetric m th-order n -dimensional tensors by $\mathbb{S}^{[m,n]}$. For any vector $x \in \mathbb{R}^n$, $\mathcal{A}x^{m-1}$ is a vector in \mathbb{R}^n with its i th component as

$$(\mathcal{A}x^{m-1})_i = \sum_{i_2, \dots, i_m=1}^n a_{ii_2 \dots i_m} x_{i_2} \cdots x_{i_m}.$$

A real symmetric tensor \mathcal{A} of order m dimension n uniquely defines a m th degree homogeneous polynomial function h with real coefficient by

$$h(x) := \mathcal{A}x^m = x^T(\mathcal{A}x^{m-1}) = \sum_{i_1, \dots, i_m=1}^n a_{i_1 \dots i_m} x_{i_1} \cdots x_{i_m}.$$

We call that the tensor \mathcal{A} is positive definite if $\mathcal{A}x^m > 0$ for all $x \neq 0$.

In 2005, Qi [29] and Lim [23] proposed the definition of eigenvalues and eigenvectors for higher order tensors, independently. Furthermore, in [5], these definitions were unified by Chang, Person and Zhang. Let \mathcal{A} and \mathcal{B} be real-valued, m th-order n -dimensional symmetric tensors. Assume further that m is even and \mathcal{B} is positive definite. we call $(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \setminus \{0\}$ is a **generalized eigenpair** of $(\mathcal{A}, \mathcal{B})$ if

$$\mathcal{A}x^{m-1} = \lambda \mathcal{B}x^{m-1}. \quad (1)$$

When the tensor \mathcal{B} is an identity tensor \mathcal{I} such that $\mathcal{I}x^{m-1} = \|x\|^{m-2}x$ for all $x \in \mathbb{R}^n$ [5], the eigenpair reduces to Z -eigenpair [29, 23]. Another special case is that when $\mathcal{B} = \mathcal{I}$ with

$$(\mathcal{I})_{i_1 i_2 \dots i_m} = \delta_{i_1 i_2 \dots i_m} = \begin{cases} 1, & \text{if } i_1 = i_2 = \dots = i_m, \\ 0, & \text{otherwise,} \end{cases}$$

the real scalar λ is called an H -eigenvalue and the real vector x is the associated H -eigenvector of the tensor \mathcal{A} [29]. In the last decade, tensor eigenproblem has received much attention in the literature [10, 14, 16, 17, 21, 22, 38, 39, 42, 43], which has numerous applications [7, 32, 33, 34].

In this paper, we consider the tensor eigenvalue complementarity problem (TEiCP):

finding a scalar $\lambda \in \mathbb{R}$, and $x \in \mathbb{R}^n \setminus \{0\}$ such that

$$\begin{aligned} x &\geq 0, \\ (\lambda \mathcal{B} - \mathcal{A})x^{m-1} &\geq 0, \\ \langle x, (\lambda \mathcal{B} - \mathcal{A})x^{m-1} \rangle &= 0, \end{aligned} \quad (2)$$

where $\mathcal{A} \in \mathbb{T}^{[m,n]}$, and $\mathcal{B} \in \mathbb{S}^{[m,n]}$ is positive definite.

The solution of TEiCP (λ, x) is called Pareto eigenpair of $(\mathcal{A}, \mathcal{B})$. In some special case, we can call it Pareto H -eigenpair or Pareto Z -eigenpair [37] if the tensor \mathcal{B} has special form as shown above in the generalized eigenpairs (1). Replacing the nonnegative cones in (2) by a closed convex cone and its dual cone, Ling, He and Qi investigated the cone

eigenvalue complementarity problem for higher-order tensor in [24]. Moreover, in [25], they studied the high-degree eigenvalue complementarity problem for tensors as a natural extension of quadratic eigenvalue complementarity problem for matrices. TEiCP is also closely related to the optimality conditions for polynomial optimization [37], a class of differential inclusions with nonconvex processes [24], and a kind of nonlinear differential dynamical system [9]. The properties of Pareto eigenvalues and their connection to polynomial optimization are studied in [37]. Recently, as a special type of nonlinear complementarity problems, the tensor complementarity problem is inspiring more and more research in the literature [2, 6, 8, 9, 13, 15, 26, 38, 39, 40]. A shifted projected power method for TEiCP was proposed in [9], in which they need an adaptive shift to force the objective to be (locally) convex to guarantee the convergence of power method. In [24], Ling, He and Qi presented a scaling-and-projection algorithm (SPA) for TEiCP. One main shortcoming of SPA is the stepsize will approach to zero as the sequence gets close to a solution of TEiCP [24]. Recently, by introducing an NCP-function, Chen and Qi [8] reformulated the TEiCP as a system of nonlinear equations. And then, they proposed a semismooth Newton method for solving the system of nonlinear equations [8].

In this paper, we will investigate two spectral projected gradient algorithms for TEiCP. The rest of this paper is organized as follows. In Section 2, some properties of the solutions of TEiCP and two optimization reformulations of TEiCP are presented. In Section 3, two spectral projected gradient algorithms are proposed. Global convergence results could be established under some suitable assumptions. We also present a shifted scaling-and-projection algorithm (SSPA) in Section 4, which is a great improvement of the original SPA method [24]. Numerical experiments are reported in Section 4 to show the efficiency of the proposed methods. Finally, we have a conclusion section.

Throughout this paper, let $\mathbb{R}_+^n = \{x \in \mathbb{R}^n : x \geq 0\}$, and $\mathbb{R}_{++}^n = \{x \in \mathbb{R}^n : x > 0\}$. Given a set $J \subseteq [n]$, the principal sub-tensor of a tensor $\mathcal{A} \in \mathbb{T}^{[m,n]}$, denoted by \mathcal{A}_J , is tensor in $\mathbb{T}^{[m,|J|]}$, such that $\mathcal{A}_J = (a_{i_1 \dots i_m})$ for all $i_1, \dots, i_m \in J$. Here, the symbol $|J|$ denotes the cardinality of J .

2 Some properties and reformulations of TEiCP

The following proposition shows the relationship between the solution of TEiCP (2) and the generalized eigenvalue problem (1).

Proposition 1 *(λ, x) is a solution of TEiCP (2) if and only if there exists a subset $I \subseteq [n]$, such that λ is a generalized eigenvalue of $(\mathcal{A}_I, \mathcal{B}_I)$ and $x_I \in \mathbb{R}_{++}^{|I|}$ is a corresponding eigenvector, and*

$$\sum_{i_2, \dots, i_m \in I} (\lambda b_{ii_2 \dots i_m} - a_{ii_2 \dots i_m}) x_{i_2} \cdots x_{i_m} \geq 0, \forall i \in \bar{I} := [n] \setminus I.$$

In such a case, the Pareto eigenvector x satisfies $x_{\bar{I}} = 0$.

This proposition was firstly presented in [37] for Pareto H-eigenpair and Pareto Z-eigenpair, and then unified by Xu and Ling for TEiCP [41].

Denote the set of solutions of (2) by $\sigma(\mathcal{A}, \mathcal{B})$, i.e.,

$$\sigma(\mathcal{A}, \mathcal{B}) = \{(\lambda, x) \in \mathbb{R} \times \mathbb{R}^n \setminus \{0\} : 0 \leq x \perp (\lambda \mathcal{B} - \mathcal{A})x^{m-1} \geq 0\}.$$

If $(\lambda, x) \in \sigma(\mathcal{A}, \mathcal{B})$, then $(\lambda, sx) \in \sigma(\mathcal{A}, \mathcal{B})$ for any $s > 0$. On the other hand, given a tensor $\mathcal{A} \in \mathbb{T}^{[m, n]}$, we know that there exists the unique semi-symmetric tensor [27] $\bar{\mathcal{A}}$ such that $\mathcal{A}x^{m-1} = \bar{\mathcal{A}}x^{m-1}$. It is clear that $\sigma(\mathcal{A}, \mathcal{B}) = \sigma(\bar{\mathcal{A}}, \mathcal{B})$. Without loss of generality, we always assume that $\mathcal{A} \in \mathbb{S}^{[m, n]}$ and just consider the solutions on the unit-sphere with $\|x\|_2 = 1$.

Proposition 2 *The symmetric TEiCP (2) is equivalent to the following optimization problem*

$$\max \lambda(x) = \frac{\mathcal{A}x^m}{\mathcal{B}x^m} \quad \text{subject to } x \in \mathbb{S}_+^{n-1} := \{x \in \mathbb{R}^n : x^T x = 1, x \geq 0\}, \quad (3)$$

in the sense that any equilibrium solution x of (3) is a solution of the symmetric TEiCP.

By some simple calculations, we can get its gradient and Hessian are as follows

$$g(x) \equiv \nabla \lambda(x) = \frac{m}{\mathcal{B}x^m} (\mathcal{A}x^{m-1} - \frac{\mathcal{A}x^m}{\mathcal{B}x^m} \mathcal{B}x^{m-1}). \quad (4)$$

and its Hessian is

$$\begin{aligned} H(x) &\equiv \nabla^2 \lambda(x) \\ &= \frac{m(m-1)\mathcal{A}x^{m-2}}{\mathcal{B}x^m} - \frac{m(m-1)\mathcal{A}x^m \mathcal{B}x^{m-2} + m^2(\mathcal{A}x^{m-1} \odot \mathcal{B}x^{m-1})}{(\mathcal{B}x^m)^2} \\ &\quad + \frac{m^2 \mathcal{A}x^m (\mathcal{B}x^{m-1} \odot \mathcal{B}x^{m-1})}{(\mathcal{B}x^m)^3}, \end{aligned}$$

where $x \odot y = xy^T + yx^T$, and $\mathcal{A}x^{m-2}$ is a matrix with its component as

$$(\mathcal{A}x^{m-2})_{i_1 i_2} = \sum_{i_3, \dots, i_m=1}^n a_{i_1 i_2 i_3 \dots i_m} x_{i_3} \cdots x_{i_m} \quad \text{for all } i_1, i_2 \in [n].$$

According to (4), we can derive that the gradient $g(x)$ is located in the tangent plane of \mathbb{S}^{n-1} at x , since

$$x^T g(x) = \frac{m}{\mathcal{B}x^m} (x^T \mathcal{A}x^{m-1} - \frac{\mathcal{A}x^m}{\mathcal{B}x^m} x^T \mathcal{B}x^{m-1}) = 0. \quad (5)$$

The Lagrangian function is $L(x, \mu, v) = \lambda(x) + \mu(x^T x - 1) + v^T x$, where $\mu \in \mathbb{R}$ and $v \in \mathbb{R}^n$ are the Lagrange multipliers. Any equilibrium solution of the nonlinear programming problem (3) satisfies the KKT conditions

$$\begin{cases} \nabla \lambda(x) + 2\mu x + v = 0, \\ v \geq 0, \\ v^T x = 0, \\ x \geq 0, \\ x^T x = 1. \end{cases}$$

Using $v^T x = 0$ and $x^T g(x) = 0$, by taking the dot product with x in the first equation, we get that $\mu = 0$. So, the first equation could be written as $v = -\nabla \lambda(x)$. Since $v \geq 0$ and \mathcal{B} is positive definite, it follows that

$$(\lambda, x) \in \{\mathbb{R} \times \mathbb{R}^n \setminus \{0\} : 0 \leq x \perp (\lambda \mathcal{B} - \mathcal{A})x^{m-1} \geq 0\},$$

i.e. any equilibrium solution x of (3) is a solution of the symmetric TEiCP (2).

Furthermore, the global maximum/minimum of $\lambda(x)$ in \mathbb{S}_+^{n-1} is corresponding to the extreme value of Pareto eigenpair of $(\mathcal{A}, \mathcal{B})$ [37, 24] if \mathcal{B} is strictly copositive, i.e., $\mathcal{B}x^m > 0$ for any $x \in \mathbb{R}_+^n \setminus \{0\}$. The concept of copositive tensor is introduced by Qi [31]. A tensor \mathcal{A} is said copositive if $\mathcal{A}x^m \geq 0$ for all $x \in \mathbb{R}_+^n \setminus \{0\}$. \mathcal{A} is copositive (strictly copositive) if and only if all of its Pareto H-eigenvalues or Z-eigenvalues are nonnegative (positive, respectively) [37].

Proposition 3 *Let $\mathcal{A}, \mathcal{B} \in \mathbb{S}^{[m,n]}$, and \mathcal{B} is copositive. Let*

$$\lambda_{TCP}^{max} = \max\{\lambda : \exists x \in \mathbb{R}^n \setminus \{0\} \text{ such that } (\lambda, x) \in \sigma(\mathcal{A}, \mathcal{B})\},$$

and

$$\lambda_{TCP}^{min} = \min\{\lambda : \exists x \in \mathbb{R}^n \setminus \{0\} \text{ such that } (\lambda, x) \in \sigma(\mathcal{A}, \mathcal{B})\}.$$

Then $\lambda_{TCP}^{max} = \max\{\lambda(x) : x \in \mathbb{S}_+^{n-1}\}$ and $\lambda_{TCP}^{min} = \min\{\lambda(x) : x \in \mathbb{S}_+^{n-1}\}$.

If both \mathcal{A} and \mathcal{B} are symmetric and strictly copositive tensors, then we can use logarithmic function as the merit function in (3). In such a case, TEiCP (2) could be reformulated to the following nonlinear optimization problem:

$$\max f(x) = \ln(\mathcal{A}x^m) - \ln(\mathcal{B}x^m) \quad \text{subject to } x \in \mathbb{S}_+^{n-1}. \quad (6)$$

Its gradient and Hessian are respectively

$$g(x) \equiv \nabla f(x) = \frac{m(\mathcal{A}x^{m-1})}{\mathcal{A}x^m} - \frac{m(\mathcal{B}x^{m-1})}{\mathcal{B}x^m}. \quad (7)$$

and

$$\begin{aligned}
H(x) &\equiv \nabla^2 f(x) \\
&= \frac{m(m-1)\mathcal{A}x^{m-2}}{\mathcal{A}x^m} - \frac{m(m-1)\mathcal{B}x^{m-2}}{\mathcal{B}x^m} + \frac{m^2\mathcal{B}x^{m-1}(\mathcal{B}x^{m-1})^T}{(\mathcal{B}x^m)^2} \\
&\quad - \frac{m^2\mathcal{A}x^{m-1}(\mathcal{A}x^{m-1})^T}{(\mathcal{A}x^m)^2}.
\end{aligned}$$

The Hessian is much simpler than that of Rayleigh quotient function in (3). If one need to use Hessian for computing Pareto eigenvalue, the logarithmic merit function may be a favorable choice.

3 Spectral projected gradient methods

In this section, the spectral projected gradient (SPG) method is applied to the non-linear programming problem (3). One main feature of SPG is the spectral choice of step length (also called BB stepsize) along the search direction, originally proposed by Barzilai and Borwein [3]. The Barzilai-Borwein method performs much better than the steepest descent gradient method or projected gradient method in practice [36, 11, 4]. Especially, when the objective function is a convex quadratic function and $n = 2$, a sequence generated by the BB method converges R -superlinearly to the global minimizer [3]. For any dimension convex quadratic function, it is still globally convergent [35] but the convergence is R -linear [12].

We firstly present the following spectral projected gradient method with monotone line search.

Algorithm 1: Spectral projected gradient (SPG1) algorithm for TEiCP

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$, an initial unit iterate $x_0 \geq 0$, parameter $\rho \in (0, 1)$. Let $\epsilon > 0$ be the tolerance of termination. Calculate gradient $g(x_0)$, $\beta_0 = 1/\|g(x_0)\|$. Set $k=0$.

Step 1: Compute $z_k = P_\Omega(x_k + \beta_k g_k)$ and the direction $d_k = z_k - x_k$.

Step 2: If $\|d_k\| = 0$ then stop: $\lambda(x) = \frac{\mathcal{A}x^m}{\mathcal{B}x^m}$ is a Pareto eigenvalue, and x is a corresponding Pareto eigenvector of TEiCP. Otherwise, set $\alpha \leftarrow 1$.

Step 3: If

$$f(x_k + \alpha d_k) \geq f(x_k) + \rho \alpha g_k^T d_k, \quad (8)$$

then define $x_{k+1} = x_k + \alpha_k d_k$, $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$. Otherwise, set $\alpha = 0.5\alpha$ and try again.

Step 4: Compute $b_k = \langle s_k, y_k \rangle$. If $b_k \leq 0$ set $\beta_{k+1} = \beta_{\max}$; else, compute $a_k = \langle s_k, s_k \rangle$ and $\beta_{k+1} = \max\{\beta_{\min}, \min\{\beta_{\max}, \frac{a_k}{b_k}\}\}$. Set $k := k + 1$ and go to **Step 1**.

Here $\Omega = \mathbb{S}_+^{n-1}$ is a close convex set. By the projection operation and the convexity of Ω , we know that for all $u \in \Omega$ and $\forall v \in \mathbb{R}^n$,

$$(v - P_\Omega(v))^T(u - P_\Omega(v)) \leq 0.$$

Set $u = x$ and $v = x + \beta g(x)$ in the above inequality, then we have

$$\beta g(x)^T[x - P_\Omega(x + \beta g(x))] + \|x - P_\Omega(x + \beta g(x))\|^2 \leq 0.$$

Let $d_\beta(x) = P_\Omega(x + \beta g(x)) - x$ with $\beta > 0$, then we have the following lemma.

Lemma 1 *For all $x \in \Omega$, $\beta \in (0, \beta_{\max}]$, we have*

$$g(x)^T d_\beta(x) \geq \frac{1}{\beta} \|d_\beta(x)\|_2^2 \geq \frac{1}{\beta_{\max}} \|d_\beta(x)\|_2^2. \quad (9)$$

From (9), we know that d_k is an ascent direction. Hence, a stepsize satisfying (8) will be found after a finite number of trials, and the SPG algorithm is well defined. When $\beta = \frac{\langle s_k, s_k \rangle}{\langle s_k, y_k \rangle}$ in $d_\beta(x_k)$, we call it spectral projected gradient (SPG). The vector $d_\beta(x^*)$ vanishes if and only if x^* is a constrained stationary point of optimization problem (3)/(6). The convergence of SPG method is established as follows. The proof is similar to that in [4].

Theorem 1 *Let $\{x_k\}$ is generated by SPG1 Algorithm. If there is a vector x_k such that $d_\beta(x_k) = 0$, then $(\lambda(x_k), x_k)$ is a solution of the symmetric TEiCP. Otherwise, any accumulation point of the sequence $\{x_k\}$ is a constrained stationary point, i.e., the sequence $\{\lambda(x_k)\}$ converges to a Pareto eigenvalue of the symmetric TEiCP.*

Proof. Let x^* be an accumulation point of $\{x_k\}$, and relabel $\{x_k\}$ a subsequence converging to x^* . According to the Proposition 2, we just need to show that x^* is a constrained stationary point of the optimization problem. Let us suppose by way of contradiction that x^* is not a constrained stationary point. So, by continuity and compactness, there exist $\delta > 0$ such that $\|d_\beta(x^*)\| \geq \delta > 0$ for all $\beta \in (0, \beta_{\max}]$. Furthermore, using the Lemma 1, we have $g(x^*)^T d_\beta(x^*) \geq \frac{\delta^2}{\beta_{\max}}$ for all $\beta \in (0, \beta_{\max}]$, which implies that for k larger enough on the subsequence that converges to x^* , $g(x_k)^T d_\beta(x_k) > c$ for all $\beta \in [\beta_{\min}, \beta_{\max}]$. Here, we can set $c = \frac{\delta^2}{2\beta_{\max}} > 0$. We consider two cases.

Firstly, assume that $\inf \alpha_k \geq \varepsilon > 0$. By continuity, for sufficiently large k , $\|d_{\beta_k}(x_k)\| \geq \delta/2$. From the line search condition (8), we have

$$f(x_k + \alpha d_k) - f(x_k) \geq \rho \alpha g_k^T d_k \geq \frac{\rho \varepsilon \delta^2}{4\beta_{\max}}.$$

Clearly, when $k \rightarrow \infty$, $f(x_k) \rightarrow \infty$, which is a contradiction. In fact, f is a continuous function and so $f(x_k) \rightarrow f(x^*)$.

Assume that $\inf \alpha_k = 0$. Since $\inf \alpha_k = 0$, there exists a subsequence $\{x_k\}_K$ such that $\lim_{k \in K} \alpha_k = 0$. In such a case, from the way α_k is chosen in (8), there exists an index \bar{k} sufficiently large such that for all $k \geq \bar{k}$, $k \in K$, for which $\alpha_k/0.5$ fails to satisfy condition (8), i.e., $f(x_k + 2\alpha d_k) - f(x_k) < 2\rho\alpha_k g_k^T d_k$. Hence,

$$\frac{f(x_k + 2\alpha d_k) - f(x_k)}{2\alpha_k} < \rho g_k^T d_k.$$

By the mean value theorem, we can rewrite this relation as

$$d_k^T g(x_k + t_k d_k) < \rho g_k^T d_k \text{ for all } k \in K, k \geq \bar{k},$$

where $t_k \in [0, 2\alpha_k]$ that goes to zero as $k \in K$ goes to infinity. Taking limits in the above inequality, we deduce that $(1 - \rho)g(x^*)^T d(x^*) \leq 0$. Since $1 - \rho > 0$ and $g_k^T d_k > 0$ for all k , then $g(x^*)^T d(x^*) = 0$. By continuity, this indicates that for k large enough on the subsequence we have that $g_k^T d_k < c/2$, which contradicts to $g_k^T d_k > c$.

Therefore, any accumulation point of the sequence $\{x_k\}$ is a constrained stationary point. By using the Proposition 2, it follows that the sequence $\{\lambda(x_k)\}$ converges to a Pareto eigenvalue of the symmetric TEiCP. \square

In the rest of this section, we would like to present the following SPG algorithm for TEiCP with curvilinear search. Its global convergence could be established similarly.

Algorithm 2: Spectral projected gradient (SPG2) algorithm for TEiCP

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$, an initial unit iterate $x_0 \geq 0$, parameter $\rho \in (0, 1)$. Let $\epsilon > 0$ be the tolerance of termination. Calculate gradient $g(x_0)$, $\beta_0 = 1/\|g(x_0)\|$. Set $k=0$.

Step 1: If $\|P_\Omega(x_k + \beta_k g_k) - x_k\| < \epsilon$, stop, declaring $\lambda(x) = \frac{\mathcal{A}x^m}{\mathcal{B}x^m}$ is a Pareto eigenvalue, and x is a corresponding Pareto eigenvector of TEiCP.

Step 2: Set $\alpha \leftarrow \beta_k$.

Step 3: Set $x_+ = P_\Omega(x_k + \alpha g_k)$. If

$$f(x_+) \geq f(x_k) + \rho\alpha g_k^T (x_+ - x_k), \tag{10}$$

then define $x_{k+1} = x_+$, $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$. Otherwise, set $\alpha = 0.5\alpha$ and try again.

Step 4: Compute $b_k = \langle s_k, y_k \rangle$. If $b_k \leq 0$ set $\beta_{k+1} = \beta_{\max}$; else, compute $a_k = \langle s_k, s_k \rangle$ and

$$\beta_{k+1} = \max\{\beta_{\min}, \min\{\beta_{\max}, \frac{a_k}{b_k}\}\}.$$

Set $k := k + 1$ and go to **Step 1**.

4 Numerical experiments

In this section, we present some numerical results to illustrate the effectiveness of the spectral projected gradient (SPG) methods, which were compared with the Scaling-and-Projection Algorithm (SPA) proposed by Ling, He and Qi [24] and the shifted projected power (SPP) method for TEiCP proposed in [9].

Both SPG1 and SPG2 are monotone ascent method. $\{x_k\}$ are always located in the feasible region Ω . In general, the merit function $f(x)$ is chosen to be the Rayleigh quotient function in (3). In the line search procedure of the SPG1 method, we used the one-dimensional quadratic interpolation to compute the stepsize α such as

$$\alpha \leftarrow \frac{-\alpha^2 g_k^T d_k}{2(f(x_k + \alpha d_k) - f(x_k) - \alpha g_k^T d_k)}.$$

In the implementation, we terminate the algorithm once

$$\|x_{k+1} - x_k\| \leq \epsilon, \text{ or } \|g(x_k)\| \leq \epsilon, \text{ or } \|\lambda_{k+1} - \lambda_k\| \leq \epsilon.$$

We accept $\epsilon = 10^{-6}$, and set the parameter $\rho = 10^{-4}$, $\beta_{\max} = \frac{1}{\|g_k\|}$ and $\beta_{\min} = \|g_k\|$. For SPP and SSPA, the parameter $\tau = 0.05$. In all numerical experiments, the maximum iterations is 500. The experiments were done on a laptop with Intel Core 2 Duo CPU with a 4GB RAM, using MATLAB R2014b, and the Tensor Toolbox [1].

We firstly describe the so-called shifted projected power (SPP) algorithm and the scaling-and-projection algorithm (SPA) as follows.

Algorithm 3: Shifted Projected Power (SPP) algorithm [9]

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$, an initial unit iterate $x_0 \geq 0$. Let $\epsilon > 0$ be the tolerance on termination. Let $\tau > 0$ be the tolerance on being positive definite.

for $k = 0, 1, \dots$ **do**

1: Compute the gradient $g(x_k) = \nabla f(x_k)$ and the Hessian $H(x_k) = \nabla^2 f(x_k)$, respectively. Let $r_k \leftarrow \max\{0, (\tau - \lambda_{\min}(H_k))/m, \nabla \hat{f}(x_k) = \nabla f(x_k) + r_k m x_k\}$.

2: Let $\nabla \hat{f}_+(x_k) = \begin{cases} 0, & \text{if } (\nabla \hat{f}(x_k))_i < 0, \\ (\nabla \hat{f}(x_k))_i, & \text{otherwise,} \end{cases}$

3: If $\|\nabla \hat{f}_+(x_k)\| \leq \epsilon$, stop. Otherwise, $x_{k+1} \leftarrow \nabla \hat{f}_+(x_k) / \|\nabla \hat{f}_+(x_k)\|$. Set $k=k+1$ and go back to Step 1.

End for

Algorithm 4: Scaling-and-Projection Algorithm (SPA) [24]

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$. For an initial point $u_0 \geq 0$, define $x_0 = u_0 / \sqrt[m]{\mathcal{B}(u_0)^m}$. Let $\epsilon > 0$ be the tolerance on termination.

for $k = 0, 1, \dots$ **do**

1: Compute $\lambda_k = \frac{\mathcal{A}(x_k)^m}{\mathcal{B}(x_k)^m}$, the gradient $g(x_k) = \nabla f(x_k) = \mathcal{A}(x_k)^{m-1} - \lambda_k \mathcal{B}(x_k)^{m-1}$.
2: If $\|g(x_k)\| \leq \epsilon$, stop. Otherwise, let $\alpha_k = \|g_k\|$, compute $u_k = P_\Omega(x_k + \alpha_k g_k)$, and $x_{k+1} = u_k / \sqrt[m]{\mathcal{B}(u_k)^m}$. Set $k=k+1$ and go back to Step 1.
End for

Since the stepsize α_k in SPA approaches to zero as the sequence $\{x_k\}$ gets close to a solution of TEiCP, as shown in [24], the number of iterations will increase significantly. In order to improve the efficiency of SPA method, they try to amplify the stepsize and proposed a modification of SPA such as $u_k = P_\Omega(x_k + s\alpha_k g_k)$ with $s \in (1, 8)$ being a constant parameter. A suitable choice s will get an improvement. But, how to choose it? Anyway, the stepsize $s\alpha_k$ also approaches to zero when the sequence $\{x_k\}$ gets close to a solution of TEiCP. When the merit function $f(x)$ is (locally) convex, this situation will be better. So, we present the following shifted SPA method, in which an adaptive shift could force the objective to be (locally) convex [19].

Algorithm 5: Shifted Scaling-and-Projection Algorithm (SSPA)

Given tensors $\mathcal{A} \in \mathbb{S}^{[m,n]}$ and $\mathcal{B} \in \mathbb{S}_+^{[m,n]}$. For an initial point $u_0 \geq 0$, define $x_0 = u_0 / \sqrt[m]{\mathcal{B}(u_0)^m}$. Compute $\lambda(x_0) = \frac{\mathcal{A}(x_0)^m}{\mathcal{B}(x_0)^m}$. Let $\epsilon > 0$ be the tolerance on termination. Let $\tau > 0$ be the tolerance on being positive definite.
for $k = 0, 1, \dots$ **do**
1: Compute $y(x_k) = \mathcal{A}(x_k)^{m-1} - \lambda_k \mathcal{B}(x_k)^{m-1}$, the Hessian $H(x_k) = \nabla^2 f(x_k)$, respectively. Let $r_k \leftarrow \max\{0, (\tau - \lambda_{\min}(H_k))/m\}$, $\hat{g}(x_k) = \nabla \hat{f}(x_k) = y(x_k) + r_k m x_k$.
2: Let $\alpha_k = \|\hat{g}(x_k)\|$, compute $u_k = P_\Omega(x_k + \alpha_k \hat{g}_k)$, and $x_{k+1} = u_k / \sqrt[m]{\mathcal{B}(u_k)^m}$, $\lambda(x_{k+1}) = \frac{\mathcal{A}(x_{k+1})^m}{\mathcal{B}(x_{k+1})^m}$.
3: If $|\lambda(x_{k+1}) - \lambda(x_k)| \leq \epsilon$, stop. Otherwise, Set $k=k+1$ and go back to Step 1.
End for

4.1 Comparison with SPA for computing Pareto Z-eigenpairs

The following example is originally from [18] and was used in evaluating the SS-HOPM algorithm in [20] and the GEAP algorithm in [19] for computing Z-eigenpairs.

Example 1 (Kofidis and Regalia [18]). Let $\mathcal{A} \in \mathbb{S}^{[4,3]}$ be the symmetric tensor defined by

$$\begin{aligned} a_{1111} &= 0.2883, & a_{1112} &= -0.0031, & a_{1113} &= 0.1973, & a_{1122} &= -0.2485, \\ a_{1223} &= 0.1862, & a_{1133} &= 0.3847, & a_{1222} &= 0.2972, & a_{1123} &= -0.2939, \\ a_{1233} &= 0.0919, & a_{1333} &= -0.3619, & a_{2222} &= 0.1241, & a_{2223} &= -0.3420, \\ & & & & a_{2233} &= 0.2127, & a_{2333} &= 0.2727, & a_{3333} &= -0.3054. \end{aligned}$$

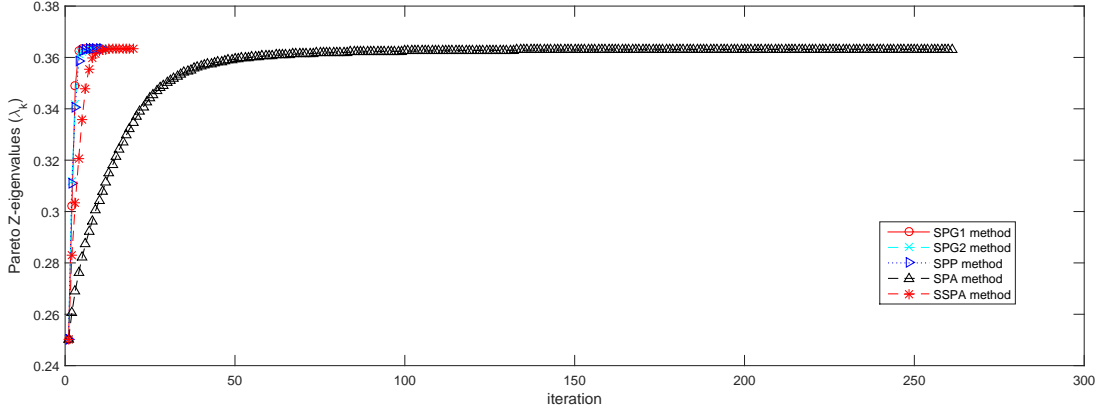


Figure 1: Comparison with SPA algorithm for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 1*, and the starting point is $x_0 = [1.0; 1.0; 1.0]$

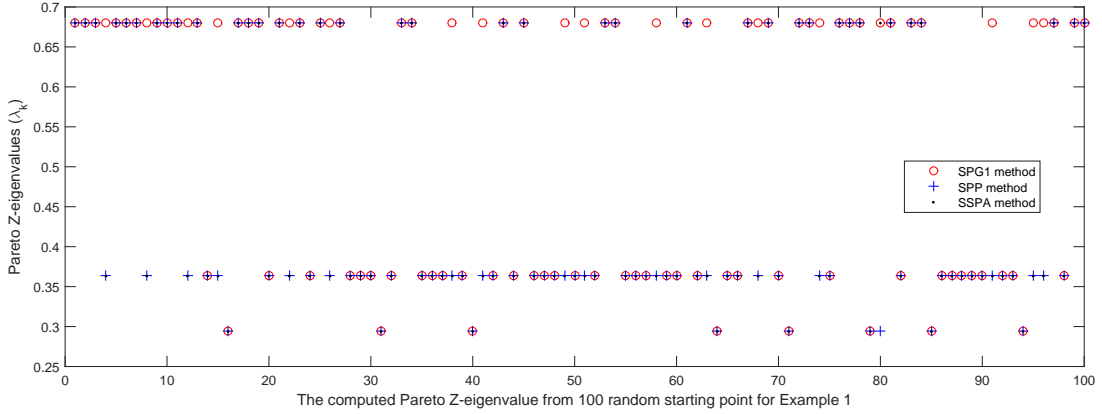


Figure 2: The computed Pareto Z-eigenvalues by SPG, SPP and SSPA in the 100 runs on the \mathcal{A} from *Example 1*.

To compare the convergence in terms of the number of iterations. Figure 1 shows the results for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 1*, and the starting point is $x_0 = [1.0; 1.0; 1.0]$. In this case, all of the SPG1, SPG2, SPP, SSPA can reach the same Pareto Z-eigenvalue 0.3633. SPG1 method just need run 9 iterations in 0.1716 seconds while SPA method need run 260 iterations in 3.3696 seconds. SPP is similar to SPG1 method in this case. SPG2 need run 13 iterations in 0.4368 seconds. As we can see, comparing with SPA method, SSPA method get a great improvement. SSPA method just need run 19 iterations in 0.2964 seconds.

Table 1. Comparison results for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 1*.

Alg.	λ	Eigenvector	Its.	Error	Time (sec.)
SPG1	0.3633	[0.2678;0.6446;0.7161]	9	5.43e-07	0.1716
SPG2	0.3633	[0.2677;0.6445;0.7162]	13	3.94e-08	0.4368
SPP	0.3633	[0.2679;0.6448;0.7158]	10	5.46e-07	0.1404
SPA	0.3632	[0.2771;0.6461;0.7112]	260	9.96e-07	3.3696
SSPA	0.3633	[0.2683;0.6449;0.7156]	19	9.00e-07	0.2964

Example 2. Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the diagonal tensor defined by $a_{iiii} = \frac{i-1}{i}$, for $i = 1, \dots, n$.

Table 2. Comparison results for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 2* with $n = 5$.

Alg.	λ	Eigenvector	Its.	Error	Time (sec.)
SPG1	0.8	[0;0;0;0;1]	3	0.0*	0.0312
SPG2	0.8	[0;0;0;0;1]	4	0.0*	0.0312
SPP	0.8	[0;2.95e-10;7.13e-09;3.22e-07;0.9999]	7	7.02e-10	0.0624
SPA	0.7999	[0.0014;0.0024;0.0037;0.0063;0.9999]	286	9.87e-07	3.4008
SSPA	0.8	[8.86e-05;1.51e-04;2.38e-04;4.31e-04;0.9999]	60	7.45e-07	0.7800

Figure 3 shows the results for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 2* with $n = 5$, and the starting point is $x_0 = [1.0; 1.0; 1.0; 1.0; 1.0]$. In this case, all of the SPG1, SPG2, SPP, SPA, SSPA can reach the largest Pareto Z-eigenvalue 0.8. SPG1 just need 3 iterations in 0.0312 seconds while SPA need run 286 iterations in 3.4008 seconds. Comparing with SPA method, SSPA method get a great improvement again. But, SSPA method is still slower than the other three methods in this case.

Example 3. Let $\mathcal{A} \in \mathbb{S}^{[4,3]}$ be the symmetric tensor defined by: Firstly, set $A = \text{tensor}(\text{zeros}(3, 3, 3, 3))$, and

$$\begin{aligned}
a_{1111} &= 1.00397, & a_{2222} &= 0.99397, & a_{3333} &= 1.00207, \\
a_{1222} &= 0.00401, & a_{2111} &= 0.00788, & a_{3111} &= 0.00001, \\
a_{3222} &= 0.00005, & a_{1333} &= 0.99603, & a_{2333} &= 1.0040,
\end{aligned}$$

and then using $A = \text{symmetrize}(A)$ to symmetrize it.

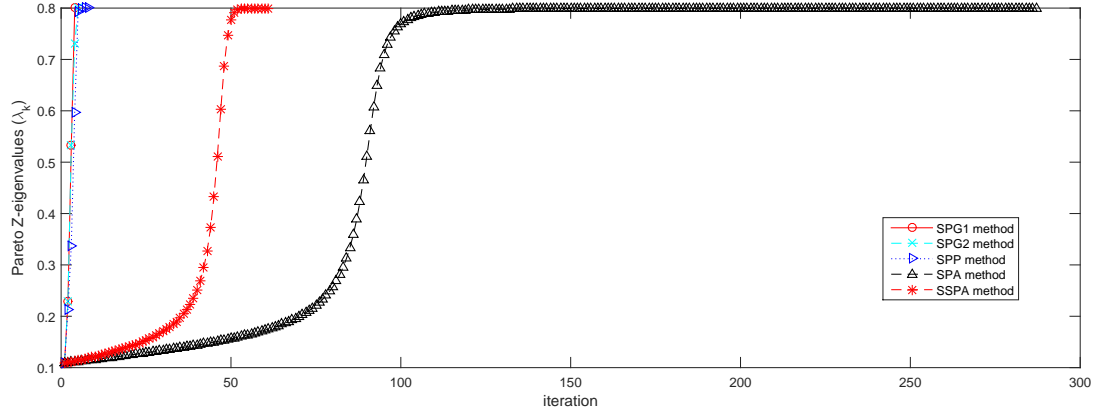


Figure 3: Comparison with SPA algorithm for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 2* with $n = 5$ and the starting point is $x_0 = [1.0; 1.0; 1.0; 1.0; 1.0]$

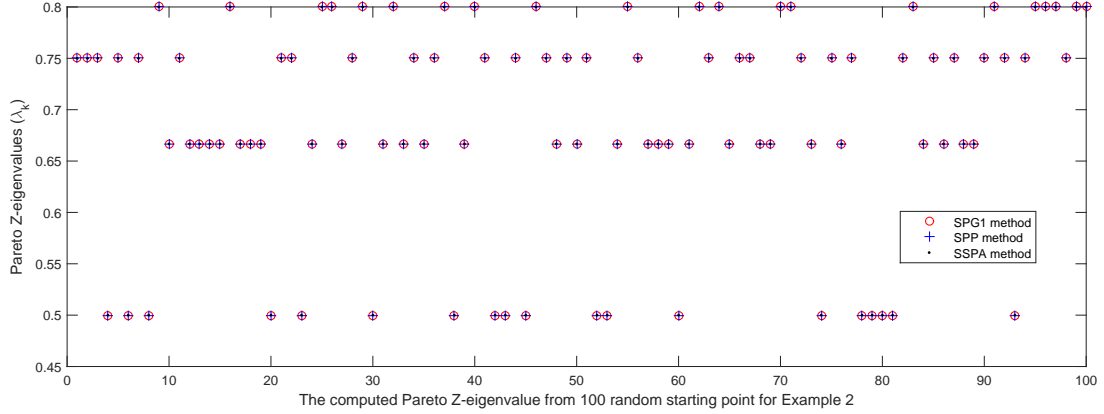


Figure 4: The computed Pareto Z-eigenvalues by SPG, SPP and SSPA in the 100 runs on the \mathcal{A} from *Example 2* with $n = 5$.

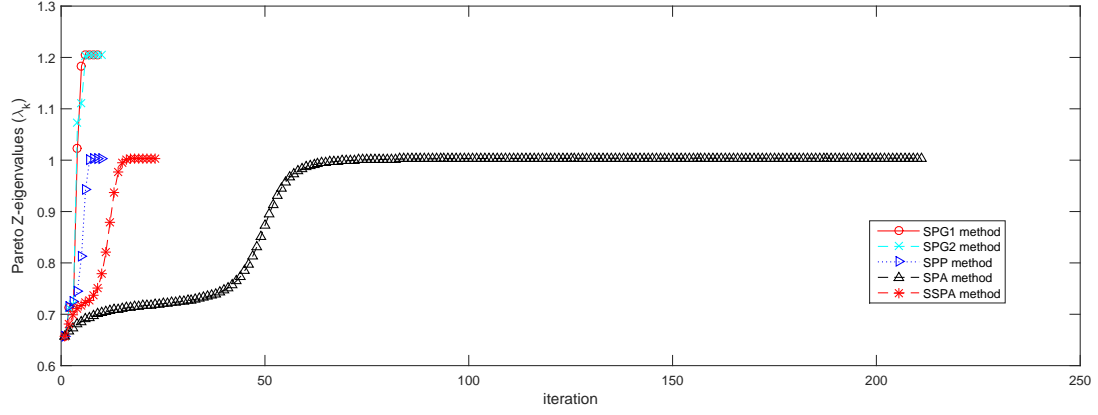


Figure 5: Comparison with SPA algorithm for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 3*, and the starting point is $x_0 = [0.9015; 0.3183; 0.5970]$

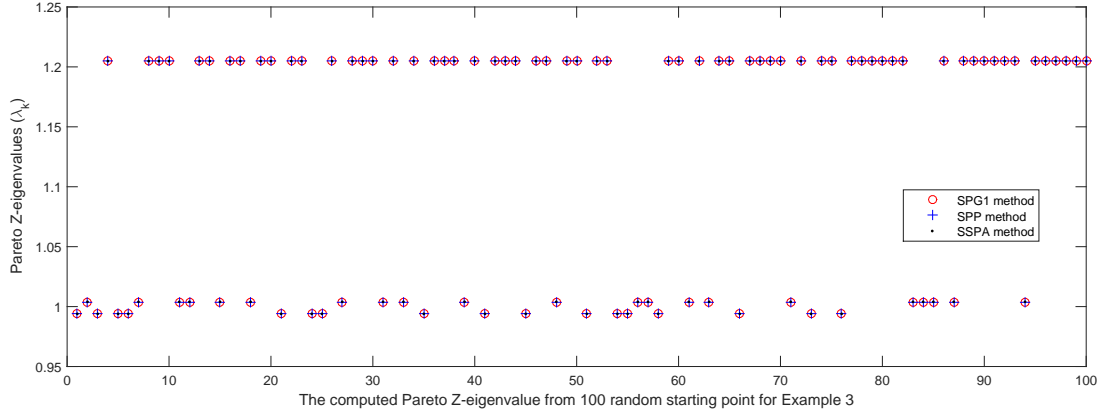


Figure 6: The computed Pareto Z-eigenvalues by SPG, SPP and SSPA in the 100 runs on the \mathcal{A} from *Example 3*.

Table 3. Comparison results for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 3*.

Alg.	λ	Eigenvector	Its.	Error	Time (sec.)
SPG1	1.2048	[0.1905;0.1920;0.9627]	8	5.25e-07	0.1248
SPG2	1.2048	[0.1902;0.1918;0.9628]	9	1.71e-07	0.2028
SPP	1.0040	[1.0;0.0020;2.72e-06]	9	7.37e-10	0.1560
SPA	1.0039	[1.0;0.0026;0.0062]	210	9.81e-07	2.6832
SSPA	1.0040	[1.0;0.0020;2.94e-04]	22	5.19e-07	0.2652

Figure 3 shows the results for computing Pareto Z-eigenvalues of \mathcal{A} from *Example 3*, and the starting point is $x_0 = [0.9015; 0.3183; 0.5970]$. In this case, both SPG1 and SPG2 reach the largest Pareto Z-eigenvalue $\lambda = 1.2048$ while the other three methods reach the Z-eigenvalue $\lambda = 1.0040$. SPG and SPP need 8/9 iterations while SPA need run 210 iterations, and SSPA need 22 iterations in this case.

We also used 100 random starting guesses, each entry selected uniformly at random from the interval $[0, 1]$, to test SPG1, SPP and SSPA, respectively. For each set of experiments, the same set of random starts was used. We listed the median number of iterations until convergence, and the average run time in the 100 experiments in Table 4. The computed Pareto Z-eigenvalues were listed in Fig.2, Fig.4, Fig.6 for Example 1,2,3, respectively. As we can see, most of time, all of the three methods can reach the same Pareto Z-eigenvalue. But for Example 1, it is seems that SPG1 method could reach the largest Pareto Z-eigenvalue with a higher probability.

Table 4. Comparison results for 100 random test on computing Pareto Z-eigenvalues of \mathcal{A} from *Ex.1, Ex.2*($n = 5$) and *Ex.3*.

Algorithm Example	SPG1		SPP		SSPA	
	Its.	Time	Its.	Time	Its.	Time
Ex. 1	7.41	0.1407	8.17	0.1259	15.61	0.2253
Ex. 2	2.11	0.0381	5.21	0.0841	37.08	0.5129
Ex. 3	4.79	0.0894	5.20	0.0853	14.30	0.2058

4.2 Comparison with SPP for computing Pareto H-eigenpairs

In this subsection, we test SPG1, SPG2, and SPP method for finding Pareto H-eigenpairs of \mathcal{A} from Examples 4-6 ($n = 5$):

Example 4 (Nie and Wang [28]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the symmetric tensor defined by

$$a_{ijkl} = \sin(i + j + k + l) \quad (1 \leq i, j, k, l \leq n).$$

Example 5 (Nie and Wang [28]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the symmetric tensor defined by

$$a_{ijkl} = \tan(i) + \tan(j) + \tan(k) + \tan(l) \quad (1 \leq i, j, k, l \leq n).$$

Example 6 (Nie and Wang [28]). Let $\mathcal{A} \in \mathbb{S}^{[4,n]}$ be the tensor defined by

$$a_{ijkl} = \frac{(-1)^i}{i} + \frac{(-1)^j}{j} + \frac{(-1)^k}{k} + \frac{(-1)^l}{l}, \quad (1 \leq i, j, k, l \leq n).$$

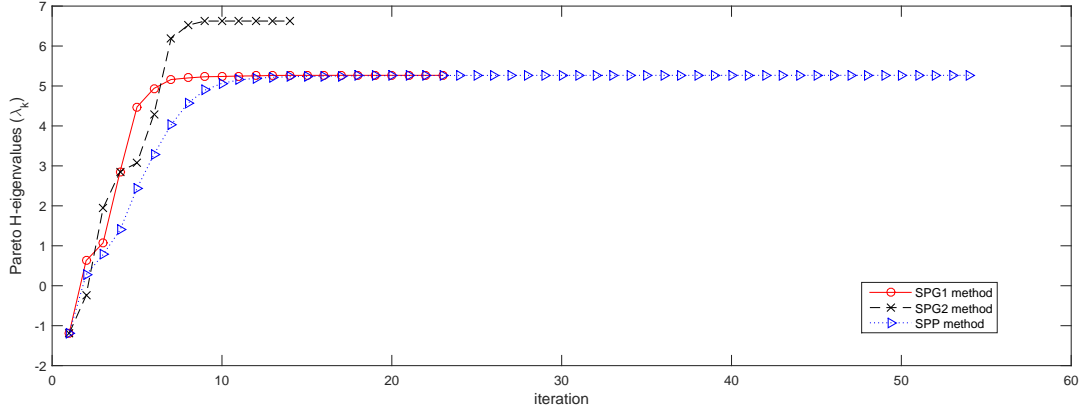


Figure 7: Comparison SPG with SPP algorithm for computing H-eigenvalues of \mathcal{A} from *Example 4* ($n=5$), and the starting point is $x_0 = [0.3319; 0.8397; 0.3717; 0.8282; 0.1765]$.

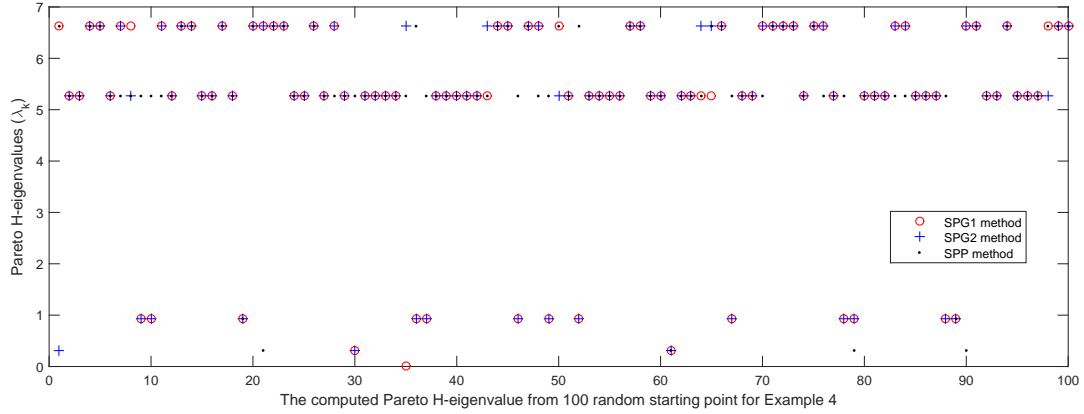


Figure 8: The computed Pareto H-eigenvalues by SPG1, SPG2, and SPP in the 100 runs on the \mathcal{A} from *Example 4*.

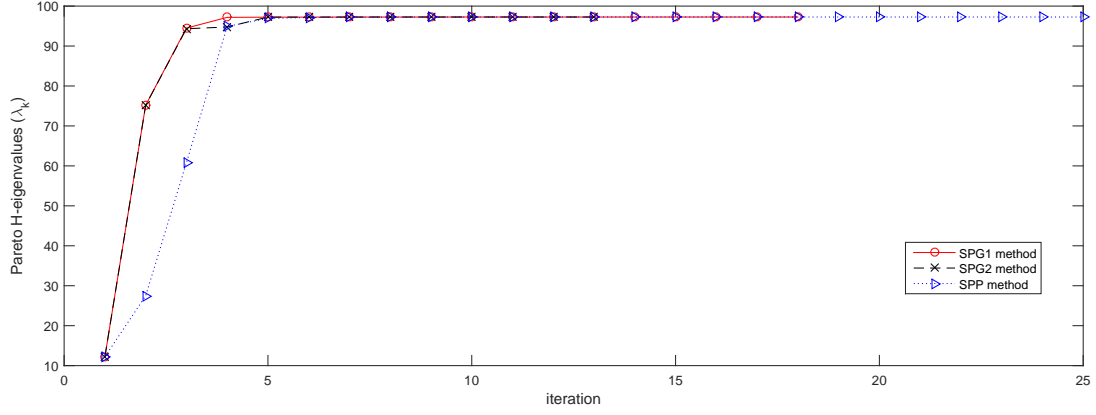


Figure 9: Comparison SPG with SPP algorithm for computing H-eigenvalues of \mathcal{A} from *Example 5* ($n=5$), and the starting point is $x_0 = [0.2291; 0.0922; 0.2409; 0.9025; 0.21734]$.

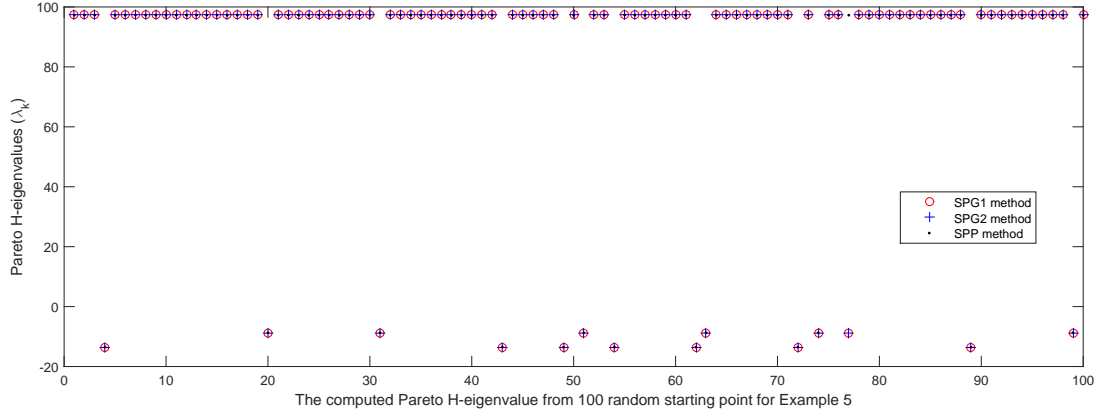


Figure 10: The computed Pareto H-eigenvalues by SPG1, SPG2, and SPP in the 100 runs on the \mathcal{A} from *Example 5*.

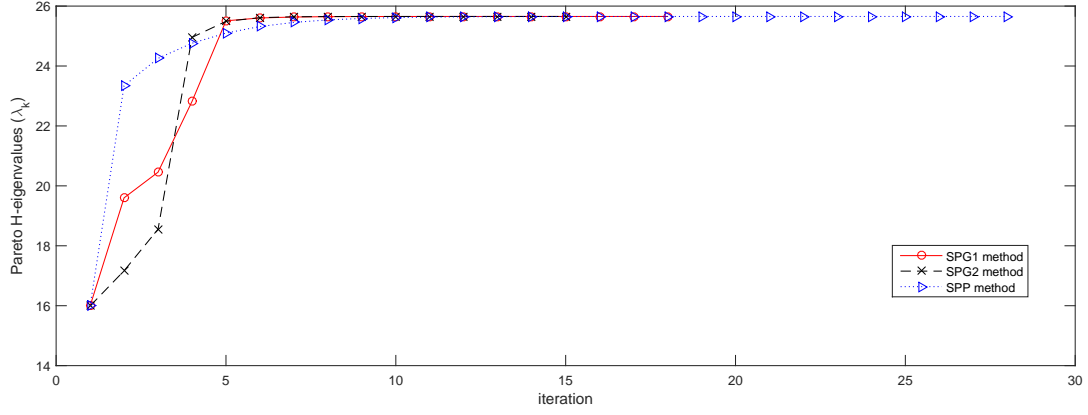


Figure 11: Comparison SPG with SPP algorithm for computing H-eigenvalues of \mathcal{A} from *Example 6* ($n=5$), and the starting point is $x_0 = [0.1846; 0.8337; 0.1696; 0.9532; 0.7225]$.

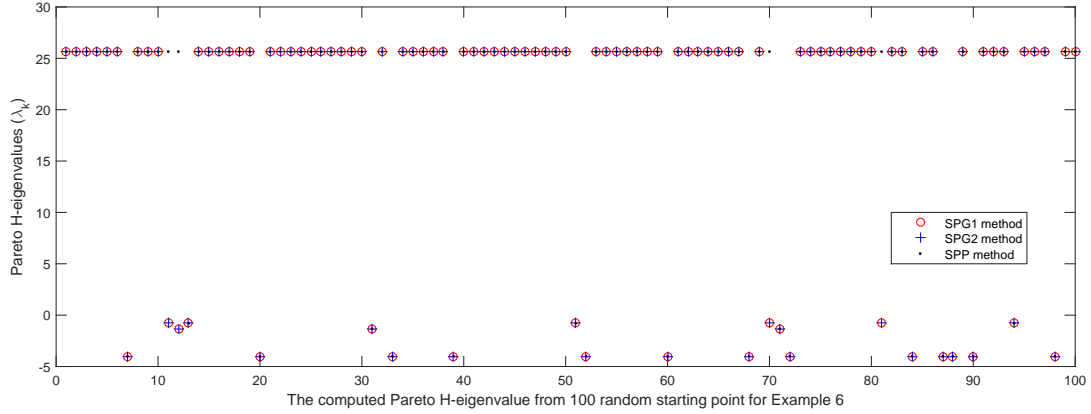


Figure 12: The computed Pareto H-eigenvalues by SPG1, SPG2, and SPP in the 100 runs on the \mathcal{A} from *Example 6*.

To compare the convergence in terms of the number of iterations. Figure 7 shows the results for computing Pareto H-eigenvalues of \mathcal{A} from *Example 4*, and the starting point is $x_0 = [0.3319; 0.8397; 0.3717; 0.8282; 0.1765]$. In this case, both of SPG1 method and SPP method can find the same Pareto H-eigenvalue 5.2664 while SPG2 method finding the Pareto H-eigenvalue 6.6255. SPG1 method need run 22 iterations in 0.4368 seconds while SPP method need run 53 iterations in 0.6240 seconds. SPG2 method just need run 13 iterations in 0.4524 seconds.

Figure 9 shows the results for computing Pareto H-eigenvalues of \mathcal{A} from *Example 5*, and the starting point is $x_0 = [0.2291; 0.0922; 0.2409; 0.9025; 0.21734]$. In this case, all of the three methods can find the largest Pareto H-eigenvalue 97.2637. SPG1 method need run 17 iterations in 0.3900 seconds while SPG2 method need run 12 iterations in 0.3820 seconds. SPP method need run 24 iterations in 0.3274 seconds.

Figure 11 shows the results for computing Pareto H-eigenvalues of \mathcal{A} from *Example 6*, and the starting point is $x_0 = [0.1846; 0.8337; 0.1696; 0.9532; 0.7225]$. In this case, all of the three methods can find the same Pareto H-eigenvalue 25.6537. SPG2 method need run 14 iterations in 0.4368 seconds. And SPG1 method need run 17 iterations in 0.2964 seconds while SPP method need run 27 iterations in 0.3276 seconds.

We also used 100 random starting guesses for finding Pareto H-eigenvalue, to test SPG1, SPG2, and SPP, respectively. For each set of experiments, the same set of random starts was used. We listed the median number of iterations until convergence, and the average run time in the 100 experiments in Table 5. The computed Pareto H-eigenvalues were listed in Fig.8, Fig.10, Fig.12 for Example 4,5,6, respectively. As we can see from the Table 5, SPP is slightly slower than SPG method. SPP method need much more iterations in general. The number of iterations of SPG2 is the least. But, SPG1 is faster than SPG2 for the test problems.

Table 5. Comparison results for 100 random test on computing Pareto H-eigenvalues of \mathcal{A} from *Examples 4-6*.

Algorithm Example	SPG1		SPG2		SPP	
	Its.	Time	Its.	Time	Its.	Time
Ex. 4	22.94	0.3861	22.51	0.5934	39.21	0.5203
Ex. 5	21.67	0.3844	13.08	0.4345	24.94	0.3354
Ex. 6	17.99	0.3151	11.09	0.3175	23.98	0.3260

5 Conclusion

In this paper, two monotone ascent spectral projected gradient algorithms were investigated for the tensor eigenvalue complementarity problem (TEiCP). We also presented a shifted scaling-and-projection algorithm, which is a great improvement of the original SPA method [24]. Numerical experiments show that spectral projected gradient methods are efficient and competitive to the shifted projected power method.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No.61262026, 11571905, 11501100), NCET Programm of the Ministry of Education (NCET 13-0738), JGZX programm of Jiangxi Province (20112BCB23027), Natural Science Foundation of Jiangxi Province (20132BAB201026), science and technology programm of Jiangxi Education Committee (LDJH12088), Program for Innovative Research Team in University of Henan Province (14IRTSTHN023).

References

- [1] B. W. Bader, T. G. Kolda and others. MATLAB Tensor Toolbox Version 2.6, Available online, February 2015. <http://www.sandia.gov/~tgkolda/TensorToolbox/>
- [2] X. Bai, Z. Huang, Y. Wang, “Global Uniqueness and Solvability for Tensor Complementarity Problems”, *arXiv:1508.06012*, Aug. 2015.
- [3] J. Barzilai and J.M. Borwein, “Two-point step size gradient methods,” *IMA J. Numer. Anal.*, 8 (1988), pp. 141-148.
- [4] E.G. Birgin, J.M. Martinez and M. Raydan, “Nonmonotone spectral projected gradient methods on convex sets”, *SIAM J. Optim.* 10 (2000), pp. 1196-1211.
- [5] K.C. Chang, K.J. Pearson and T. Zhang, “On eigenvalue problems of real symmetric tensors”, *J. Math. Anal. Appl.*, 350 (2009) 416-422.
- [6] M. Che, L. Qi, Y. Wei, “Positive Definite Tensors to Nonlinear Complementarity Problems”, *arXiv:1501.02546*, Jan. 2015.
- [7] Y. Chen, Y. Dai, D. Han, and W. Sun, “Positive semidefinite generalized diffusion tensor imaging via quadratic semidefinite programming”, *SIAM J. Imaging Sci.*, 6 (2013) 1531-1552.
- [8] Z. Chen and L. Qi, “A semismooth Newton method for tensor eigenvalue complementarity problem”, *arXiv: 1510.08570v1*, October 2015.
- [9] Z. Chen, Q. Yang and L. Ye, “Generalized eigenvalue complementarity problem for tensors”, *arXiv: 1505.02494v1*, May 2015.
- [10] C. Cui, Y. Dai and J. Nie, “All real eigenvalues of symmetric tensors”, *SIAM J. Matrix Anal. Appl.*, 35 (2014) 1582-1601.
- [11] Y.H. Dai and R. Fletcher, “Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming”, *Numerische Mathematik*, 100 (2005), pp. 21-47.

- [12] Y.H. Dai and L.Z. Liao, “ R -linear convergence of the Barzilai and Borwein gradient method,” *IMA J. Numer. Anal.*, 22 (2002), pp. 1-10.
- [13] W. Ding, Z. Luo, L. Qi, “P-Tensors, P_0 -Tensors, and Tensor Complementarity Problem”, *arXiv:1507.06731*, July 2015.
- [14] W. Ding and Y. Wei, “Generalized tensor eigenvalue problems”, *SIAM J. Matrix Anal. Appl.*, 36 (2015) 1073-1099.
- [15] M. S. Gowda, Z. Luo, L. Qi, N. Xiu, “Z-tensors and complementarity problems”, *arXiv:1510.07933*, Oct. 2015.
- [16] C. Hao, C. Cui and Y. Dai, “A sequential subspace projection method for extreme Z-eigenvalues of supersymmetric tensors”, *Numer. Linear Algebr. Appl.*, 22 (2015) 283-298.
- [17] C. Hao, C. Cui and Y. Dai, “A feasible trust-region method for calculating extreme Z-eigenvalues of symmetric tensors”, *Pacific J. Optim.*, 11 (2015) 291-307.
- [18] E. Kofidis and P. Regalia, “On the best rank-1 approximation of higher-order supersymmetric tensors”, *SIAM J. Matrix Anal. Appl.*, 23 (2002) 863-884.
- [19] T. G. Kolda and J. R. Mayo, “An adaptive shifted power methods for computing generalized tensor eigenpairs”, *SIAM J. Matrix Anal. Appl.*, 35 (2014) 1563-1581.
- [20] T. G. Kolda and J. R. Mayo, “Shifted power method for computing tensor eigenpairs”, *SIAM J. Matrix Anal. Appl.*, 32 (2011) 1095-1124.
- [21] G. Li, L. Qi, G. Yu, “The Z-eigenvalues of a symmetric tensor and its application to spectral hypergraph theory”, *Numerical Linear Algebra with Applications*, 20 (2013) 1001-1029.
- [22] G. Li, L. Qi, G. Yu, “Semismoothness of the maximum eigenvalue function of a symmetric tensor and its application”, *Linear Algebra and its Applications*, 438 (2013) 813-833.
- [23] L.H. Lim, “ Singular values and eigenvalues of tensors, A variational approach”, Proc. 1st IEEE International workshop on computational advances of multi-tensor adaptive processing, 2005, 129-132.
- [24] C. Ling, H. He and L. Qi, “On the cone eigenvalue complementarity problem for higher-order tensors”, *Computational Optimization and Applications*, to appear.
- [25] C. Ling, H. He and L. Qi, “Higher-degree eigenvalue complementarity problem for tensors”, *Computational Optimization and Applications*, to appear.

- [26] Z. Luo, L. Qi, N. Xiu, “The Sparsest Solutions to Z-Tensor Complementarity Problems”, *arXiv:arXiv:1505.00993*, May 2015.
- [27] Q. Ni and L. Qi, “A quadratically convergent algorithm for finding the largest eigenvalue of a nonnegative homogeneous polynomial map”, *Journal of Global Optimization*, 61 (2015), pp. 627-641.
- [28] J. Nie, L. Wang, “Semidefinite relaxations for best rank-1 tensor approximations,” *SIAM Journal on Matrix Analysis and Applications* 35 (2014), pp. 1155-1179.
- [29] L. Qi, “Eigenvalues of a real supersymmetric tensor”, *J. Symbolic Computation*, 40 (2005) 1302-1324.
- [30] L. Qi, “Eigenvalues and invariants of tensor”, *J. Math. Anal. Appl.*, 325 (2007) 1363-1377.
- [31] L. Qi, “Symmetric nonnegative tensors and copositive tensors”, *Linear Algebra and its Applications* 439 (2013) 228-238.
- [32] L. Qi, Y. Wang and E.X. Wu, “D-Eigenvalues of diffusion kurtosis tensors”, *Journal of Computational and Applied Mathematics*, 221 (2008) 150-157.
- [33] L. Qi, G. Yu and E.X. Wu, “Higher order positive semi-definite diffusion tensor imaging”, *SIAM Journal on Imaging Sciences*, 3 (2010) 416-433.
- [34] L. Qi, G. Yu and Y. Xu, “Nonnegative diffusion orientation distribution function”, *Journal of Mathematical Imaging and Vision*, 45 (2013) 103-113.
- [35] M. Raydan, “On the Barzilai and Borwein choice of steplength for the gradient method” ,*IMA J. Numer. Anal.*, 13 (1993), pp. 321-326.
- [36] M. Raydan, “The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem”, *SIAM J. Optim.*, 7 (1997) pp. 26-33.
- [37] Y. Song and L. Qi, “Eigenvalue analysis of constrained minimization problem for homogeneous polynomial”, *Journal of Global Optimization*, to appear.
- [38] Y. Song and L. Qi, “Tensor Complementarity Problem and Semi-positive Tensors”, *Journal of Optimization Theory and Applications*, to appear.
- [39] Y. Song and G. Yu, “Properties of Solution Set of Tensor Complementarity Problem”, *arXiv:1508.00069v2*, Aug. 2015.
- [40] Y. Wang, Z. Huang, X. Bai, “Exceptionally Regular Tensors and Tensor Complementarity Problems”, *arXiv:1508.06422*, Aug. 2015.

- [41] F. Xu, C. Ling, “Some properties on Pareto-eigenvalues of higher-order tensors”, *Operations Research Transactions*, 19 (2015) 34-41.
- [42] L. Yang, Q. Yang, X. Zhao, “Quadratic third-order tensor optimization problem with quadratic constraints”, *Statistics, Optimization and Information Computing*, 2 (2014) 130-146.
- [43] Y. Yang, Q. Yang, “Further results for Perron-Frobenius Theorem for nonnegative tensors”, *SIAM Journal on Matrix Analysis and Applications* 31 (2010) 2517-2530.